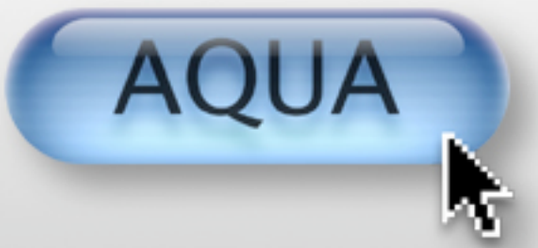
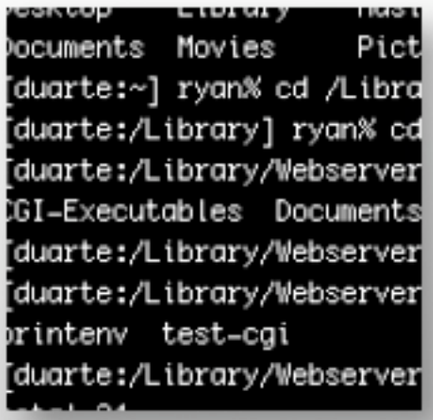
What You Need to Know About Fonts

Session 406





What You Need to Know About Fonts

Xavier Legros
Mac OS X Evangelist, Apple Worldwide Developer Relations
xavier@apple.com

What You Need to Know About Fonts

Robin Mikawa
Manager, Apple Type Engineering
rmikawa@apple.com

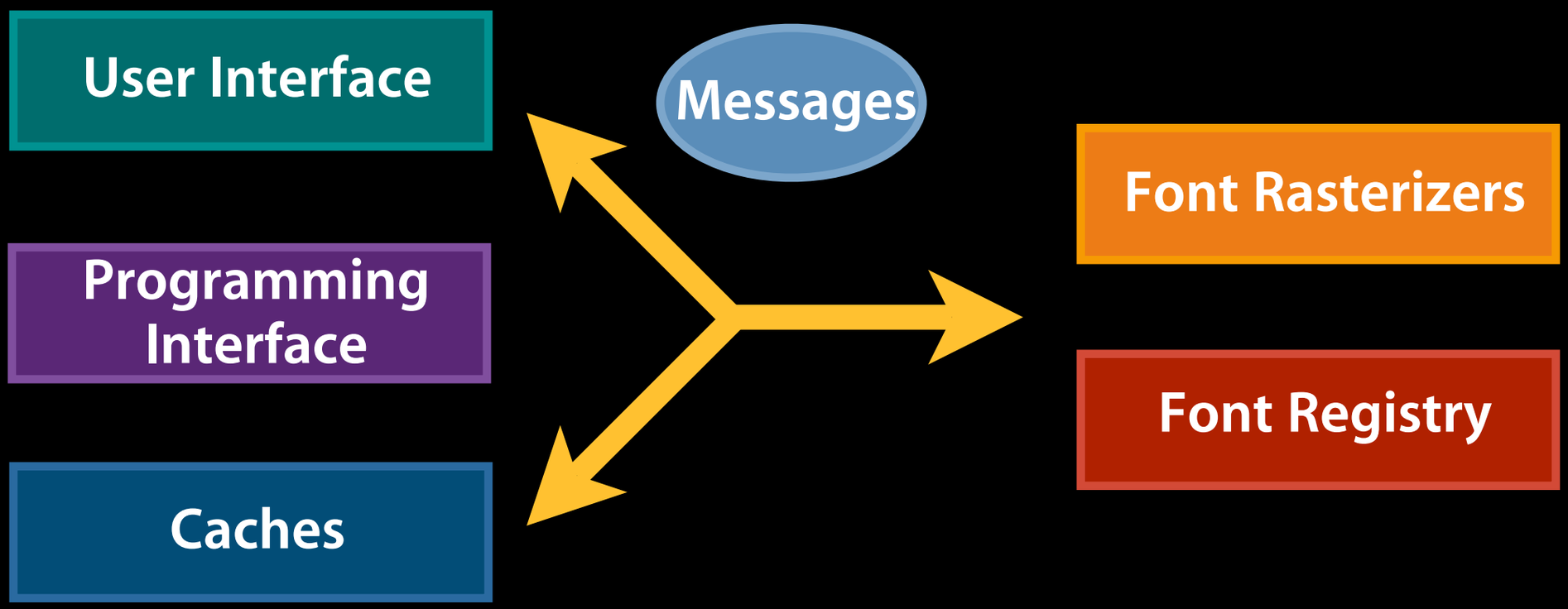
Introduction

- Why are fonts important?
- What will you learn from this session?
  - What does the operating system do?
  - What do you need to do?

What Does the Operating System Do?

- Maintain the font registry
- Rasterize and convert font data
- Handle messages, notifications, and queries
- Implement caches and performance optimizations
- Implement the programming interface
- Implement the standard user interface

What Does the Operating System Do?

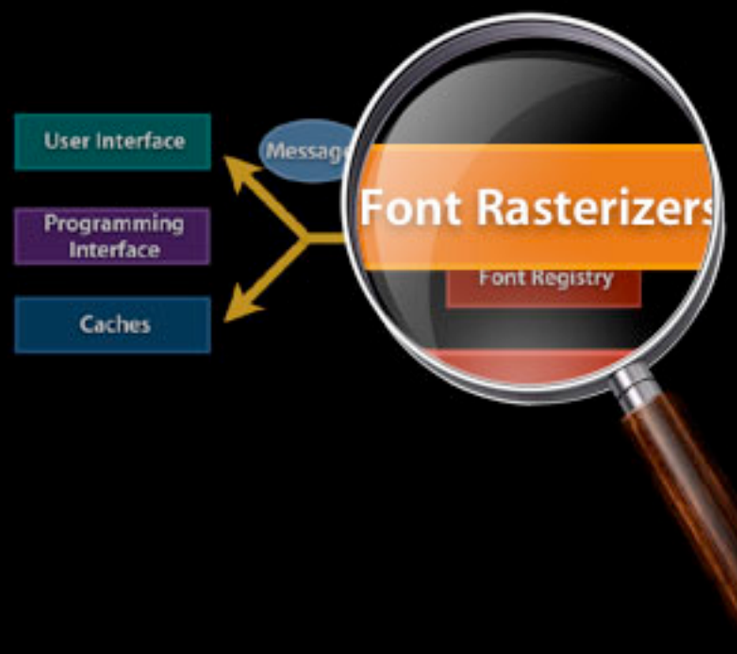


Font Registry

- Maintain information on where the fonts are stored
  - Standard font directories
  - Multiple users and access privileges
- Maintain information on state of the fonts
- Manage all supported font file formats

Font Rasterizers

- Convert font information to graphics data
- Generate font data for printing and PDF files
- Process all supported data formats
  - TrueType (Macintosh and Windows)
  - PostScript (Type 1 and Compact Font Format)
  - Bitmap (QuickDraw-based applications only)

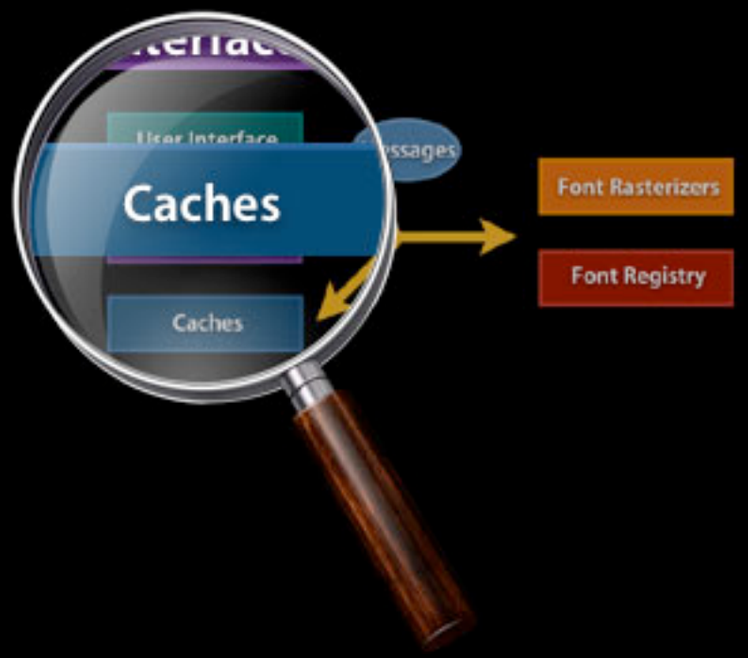


Messages

- Interprocess communications with font server
- Implement the programming interface
- Notify processes of changes to font registry
- Notify operating system of changes to font directories
- Generated by processes querying for information on missing fonts

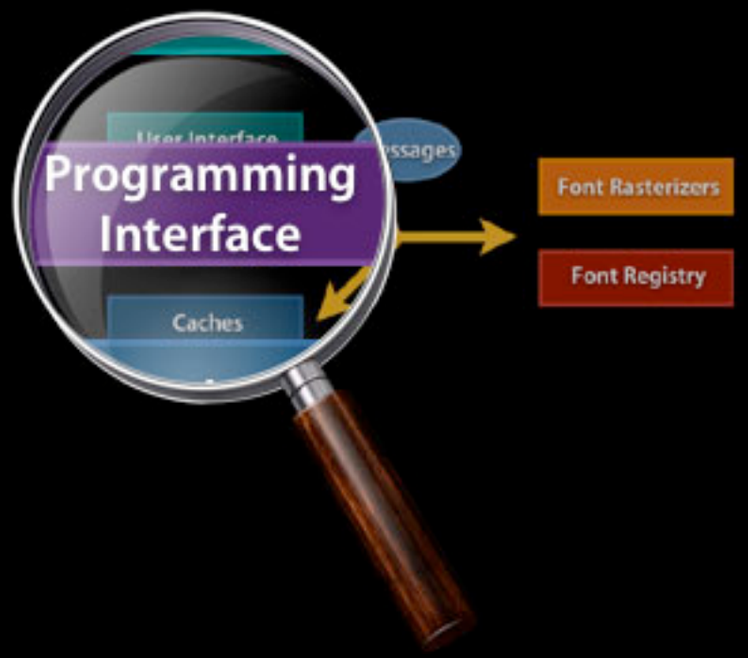
Caches

- Optimize performance for commonly used functions
- Reduce file system usage
- Reduce font processing
- Reduce messaging



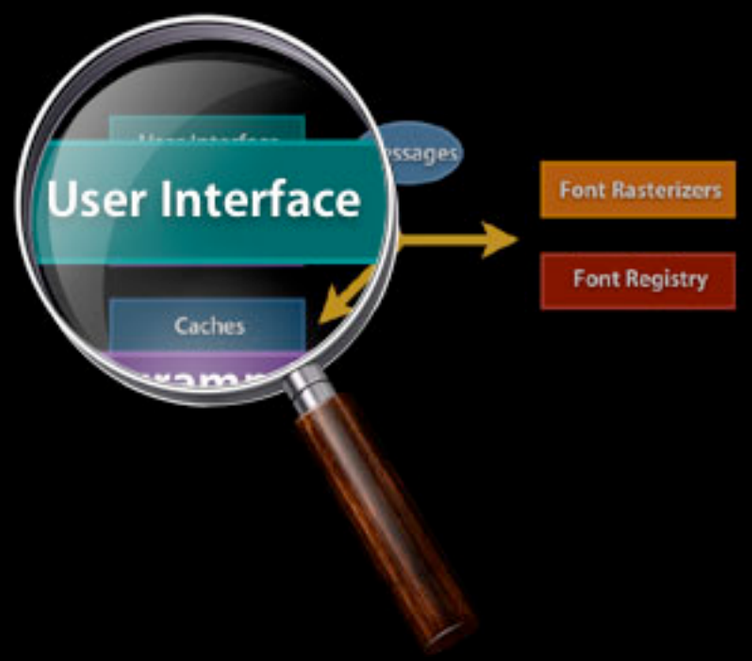
Programming Interface

- Manage the contents of the registry
- Access font data from the registry
- Access glyph data from the rasterizers
- Handle notifications and queries



User Interface

- Change font and style information
- Access glyph information on a font
- Access typographic features of a font
- Manage collections of fonts



Demo

Nathan Taylor
Software Engineer, Apple Type Engineering

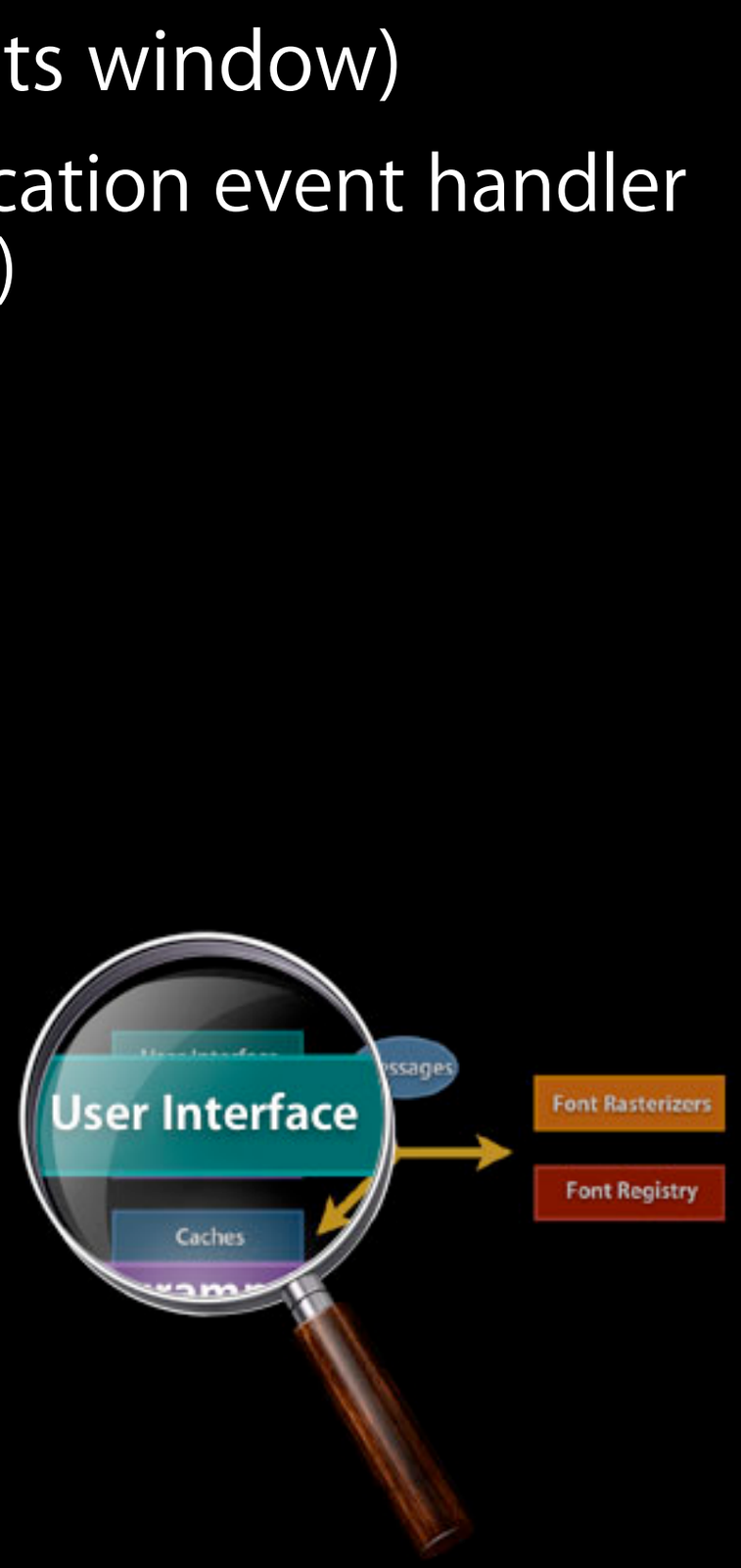
http://developer.apple.com/wwdc2003/urls.html

What Do You Need To Do?

- ✅ Adopt the standard user interface
- ✅ Handle font notifications
- Exchange data with Cocoa and Carbon
- Manage the contents of the registry
- Access font data from the registry
- Access glyph data from the rasterizers

Adopting the Standard User Interface

- Show and hide the font panel (Fonts window)
  - Supported through standard application event handler (kHICommandShowHideFontPanel)
- Handle font selection events
- Set the font selection
- Handle user focus



Handling Font Selection Events

- Handle font selection events from the font panel
  - Handle the font selection event type (kEventClassFont, kEventFontSelection)
  - Obtain information on the selection from the event parameters
    - Typeface family, typeface, size
    - Typographic features
    - Text style information
    - Multiple Master variation information

Setting the Font Selection

- Set the selection in the font panel to reflect the state of the active document window

```
myHIObjectTarget = (HIObjectRef)
            GetWindowEventTarget(thisWindow);

SetFontInfoForSelection(kFontSelectionATSUIType,
            myStyleCount,
            myStyleArray,
            myHIObjectTarget);
```

Handling User Focus

- Track user focus for the user interface elements to which you want font selection events to be sent

```
if (GetEventKind(event) == kEventWindowFocusRelinquish)
    SetFontInfoForSelection(kFontSelectionATSUIType,
    0, NULL, NULL);

if (GetEventKind(event) == kEventWindowFocusAcquired)
{
    status = GetEventParameter(event,
            kEventParamDirectObject,
            typeWindowRef, NULL,
            sizeof(WindowRef), NULL,
            &thisWindow);
    ...
}
```

What Do You Need to Do?

- Adopt the standard user interface
- ✅ Handle font notifications
- Exchange data with Cocoa and Carbon
- Manage the contents of the registry
- Access font data from the registry
- Access glyph data from the rasterizers

Handling Font Notifications

- Use notifications as an efficient way for your application to keep up-to-date on changes to the font registry
- Send notifications to the operating system when your application makes changes to a font directory

Handling Font Notifications

- Implement a callback function to handle the notification and register it with the operating system

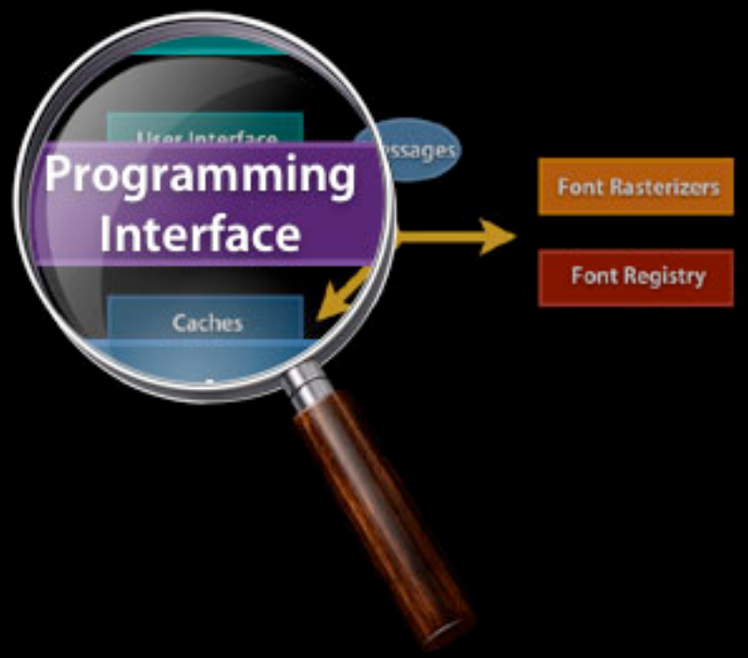
```
static void MyNotificationCallback(
    ATSFontNotificationInfoRef myInfoRef,
    void * myRefConData) { ... }

status = ATSFontNotificationSubscribe(
            MyNotificationCallback,
            kATSFontNotifyOptionDefault,
            myRefConData,
            &myNotifyRef);

status = ATSFontNotificationUnsubscribe(&myNotifyRef);
```

What Do You Need to Do?

- Adopt the standard user interface
- Handle font notifications
- Exchange data with Cocoa and Carbon
- Manage the contents of the registry
- Access font data from the registry
- Access glyph data from the rasterizers

Exchanging Data with Cocoa

- Use PostScript name (Quartz, Cocoa)
- Use conversion functions
  - NSFont fontWithName:size:
  - NSFontDescriptor fontDescriptorWithName:size:
  - NSFontDescriptor fontDescriptorWithFontAttributes:



Exchanging Data With Carbon

- Use font family name and style (ATSUI, QuickDraw)
- Use conversion functions
  - Handle differences in font family references
  - Handle differences in font references
- Avoid type casting
- Avoid implicit type promotion

What Do You Need to Do?

- Adopt the standard user interface
- Handle font notifications
- Exchange data with Cocoa and Carbon
- Manage the contents of the registry
- Access font data from the registry
- Access glyph data from the rasterizers

Managing the Font Registry

- Control which fonts are available to users by activating and deactivating fonts
- Add, remove, and modify the contents of font collections

Activating and Deactivating Fonts

- Control which fonts are available to users by activating and deactivating fonts

```
status = ATSFontActivateFromFileSpecification(
            &myFontFileSpec,
            kATSFontContextLocal,
            kATSFontFormatUnspecified,
            NULL,
            kATSOptionFlagsDefault,
            &myFontContainer);

status = ATSFontDeactivate(myFontContainer,
            NULL,
            kATSOptionFlagsDefault);
```

Modifying Font Collections

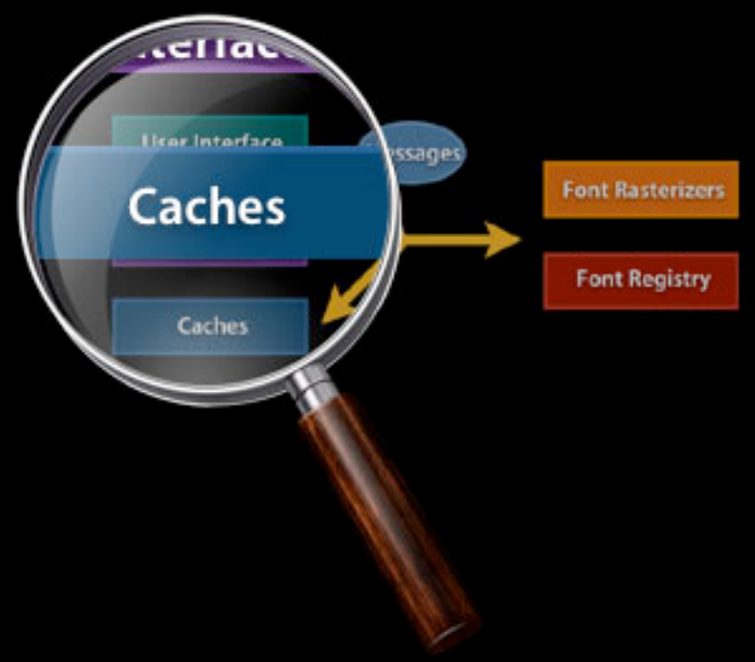
- Use methods from the NSFontManager class of the Cocoa text system to modify font collections
- Use C-callable wrapper functions defined for Carbon
- Add and remove font collections by name
- Modify the contents of font collections through font descriptors

What Do You Need to Do?

- Adopt the standard user interface
- Handle font notifications
- Exchange data with Cocoa and Carbon
- Manage the contents of the registry
- Access font data from the registry
- Access glyph data from the rasterizers

Accessing Font Data

- Use high-level functions from the ATS font system to access cached data
  - ATSFontGetPostScriptName
  - ATSFontGetFileSpecification
- Use low-level functions to access data directly from the file system
- Use enumeration functions to gather information on all fonts in the registry



Accessing Font Data

- Use low-level functions from the ATS font system to access data directly from the file system

```
status = ATSFontGetTable(fontRef, tableTag, 0, 0, NULL,
            &tableLength);

if (status == noErr)
    status = ATSFontGetTable(fontRef, tableTag, 0,
                min(tableLength, sizeof(tableBuffer)),
                &tableBuffer,
                &tableLength);
```

Enumerating Fonts and Font Families

- Use an iterator to enumerate fonts and font families tracked by the font registry

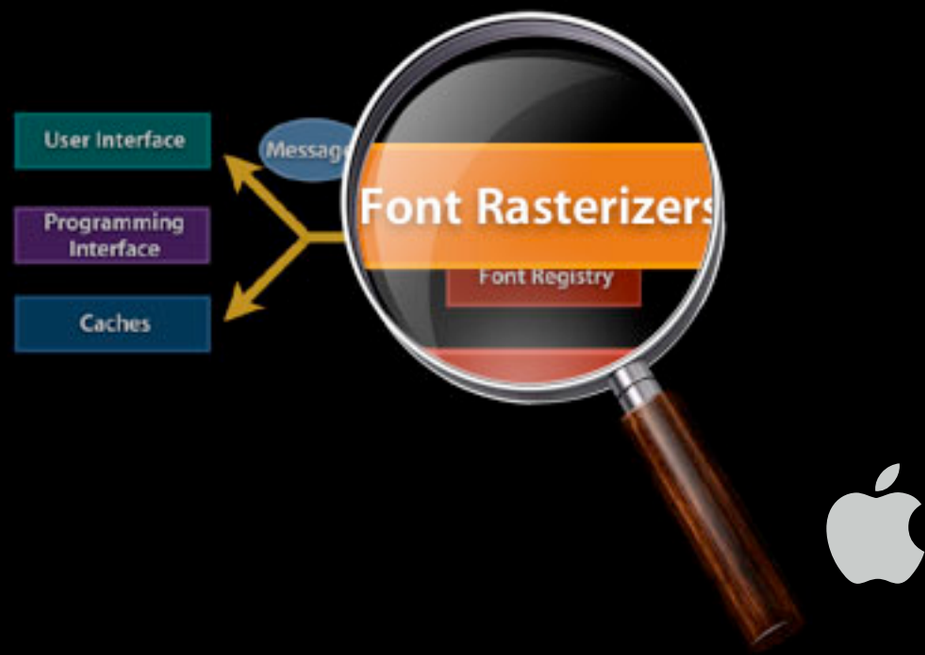
```
status = ATSFontIteratorCreate(
            kATSFontContextGlobal,
            &myFontFilter,
            &myRefConData,
            kATSOptionFlagsUnrestrictedScope,
            &myFontIterator);

while (status == noErr)
    status = ATSFontIteratorNext(
                &myFontIterator,
                &myFontRef);

status = ATSFontIteratorRelease(&myFontIterator);
```

What Do You Need to Do?

- Adopt the standard user interface
- Handle font notifications
- Exchange data with Cocoa and Carbon
- Manage the contents of the registry
- Access font data from the registry
- Access glyph data from the rasterizers

Accessing Glyph Data

- Use functions from the ATSUI text system to access glyph path data from the rasterizers
  - ATSUGetQuadraticGlyphPaths
  - ATSUGetCubicGlyphPaths
- Determine the glyph data format of the font
- Implement callback functions to process the line and curve segments that define the shape of the glyphs



Conclusion

- The operating system handles the complexity of working with fonts
- You can implement support for technologies that enhance the user experience for fonts
  - Adopt the standard user interface
  - Handle font notifications

Roadmap

404 - Unicode for Japanese, Chinese, and Everything Else!

Come learn how Unicode can give your application new international muscle

Russian Hill
Tues., 2:00pm

427 - Cocoa Text

The text system has always been one of the central components of Cocoa, but this year it's getting some exciting new features that will make it more useful and powerful than ever

Presidio
Fri., 5:00pm

Who to Contact

Xavier Legros
Mac OS X Evangelist, Apple Worldwide Developer Relations
xavier@apple.com

http://developer.apple.com/wwdc2003/urls.html

Reference Library

Documentation

Inside Mac OS X: Managing Fonts

http://developer.apple.com/techpubs/macosx/Carbon/text/ATSTypes/ATS_Concepts/index.html

Apple Type Services for Fonts Reference

http://developer.apple.com/techpubs/macosx/Carbon/text/ATSTypes/ATS/atsfontsref_Reference/index.html

Fonts Window Services Reference

http://developer.apple.com/techpubs/macosx/Carbon/HumanInterfaceToolbox/FontWindowServices/FontsWindow/fontswin_refchap/index.html

Reference Library

Related Documentation

Inside Mac OS X: Cocoa Text System Architecture

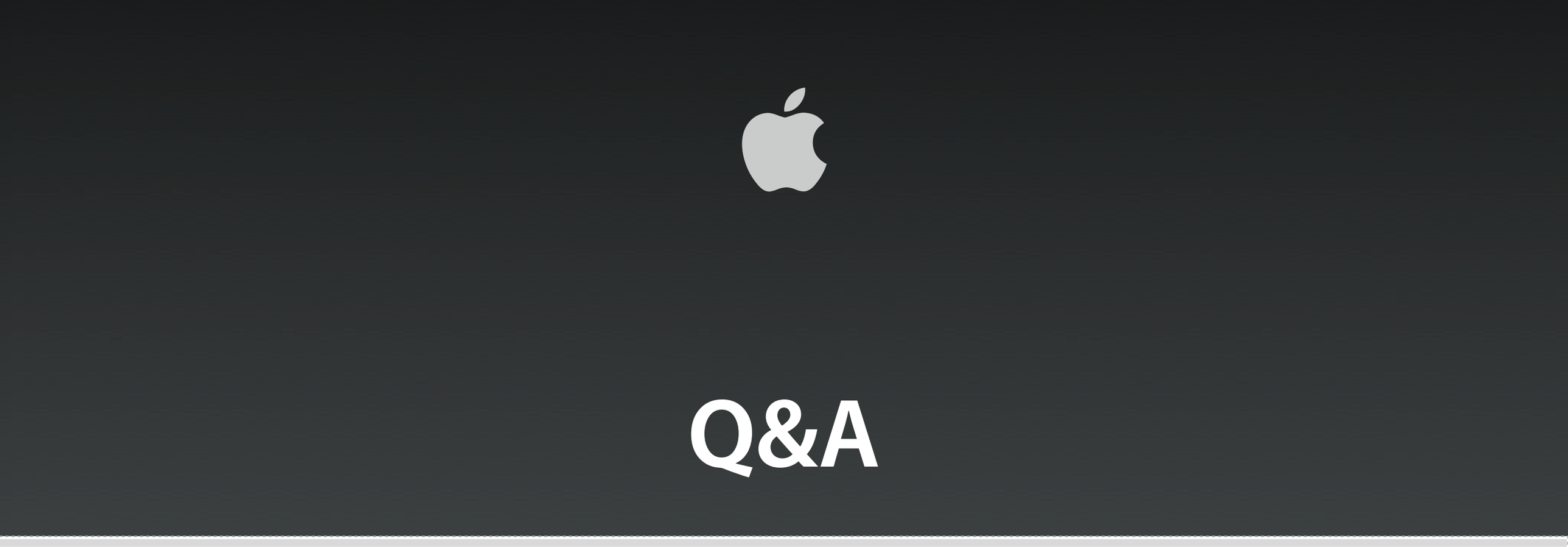
http://developer.apple.com/techpubs/macosx/Cocoa/TasksAndConcepts/ProgrammingTopics/TextArchitecture/index.html

Inside Mac OS X: Rendering Unicode Text With ATSUI

http://developer.apple.com/techpubs/macosx/Carbon/text/ATSUI/ATSUI_Concepts/index.html

Inside Mac OS X: Managing Fonts With the Font Manager

http://developer.apple.com/techpubs/macosx/Carbon/text/FontManager/Managing_FontManager/index.html



Q&A

Xavier Legros
Mac OS X Evangelist, Apple Worldwide Developer Relations
xavier@apple.com

http://developer.apple.com/wwdc2003/urls.html

WWDC 2003

Worldwide Developers Conference